

Llamadas al Sistema

Para entender y estudiar las llamadas al sistema debemos tener en claro el concepto de Kernel:

Kernel. Es el componente central de un sistema operativo. Actúa como interfaz entre las aplicaciones del usuario y el hardware. Su objetivo es administrar la comunicación entre las aplicaciones (a nivel del usuario) y el hardware (CPU, memoria, disco duro, etc.). Sus principales tareas son:

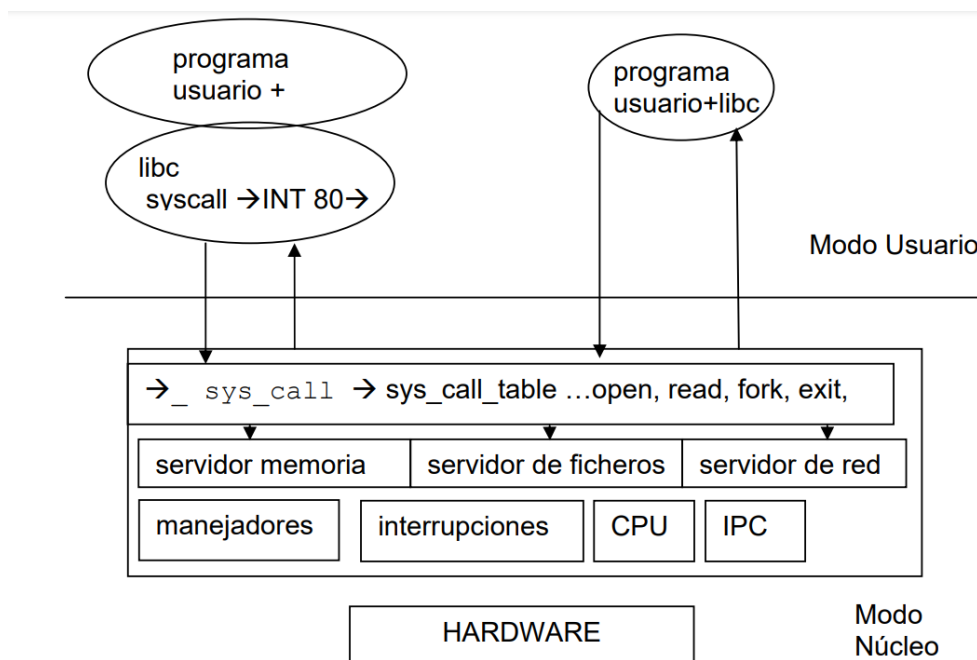
- Administrar los procesos,
- Administrar los dispositivos,
- Administrar la memoria,
- Manejar las interrupciones,
- Realizar las comunicaciones de entrada/salida,
- Administrar los archivos del sistema, etc.

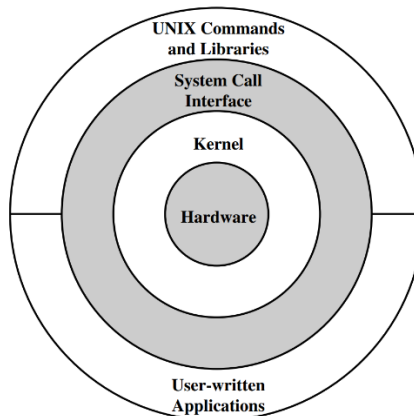
La diferencia entre el S.O. y el kernel es que éste último es el “corazón” del S.O. Si al kernel le agregamos utilidades y aplicaciones, entonces se convierte en lo que conocemos como S.O.

$$S.O. = kernel \cup \text{espacio del usuario}$$

Llamadas al sistema. Es un pequeño grupo de funciones que permiten controlar archivos y dispositivos (realizar todas las tareas anteriormente mencionadas). Estas son proporcionadas precisamente por el *kernel*. En el *kernel* o núcleo se encuentran también los controladores (interfaces de bajo nivel para controlar el hardware).

Una de las llamadas del sistema más utilizada es *fcntl*, esta llamada al sistema permite el manejo de archivos. Las siguientes imágenes permiten posicionar visualmente al *kernel* y a las llamadas de sistema (interfaz entre el *kernel* y las aplicaciones de los usuarios).





Tomando en cuenta las tareas que realizan las llamadas al sistema tenemos varios.

- Control de procesos. Dentro de este tipo tenemos subtareas que realizan las llamadas al sistema: terminar o abortar procesos; cargar y ejecutar programas; crear procesos, obtener datos de los procesos; hacer que los procesos se pongan en estado de espera (*wait*), y asignarles memoria a los procesos
- Administración de archivos: crear, borrar, abrir o cerrar archivos. También leerlos, escribir sobre ellos, obtener o asignar atributos a los archivos (información acerca de las características de los archivos).
- Administran los dispositivos: leer desde un dispositivo, escribir en ese dispositivo, adjuntar o despegar (*detach*) dispositivos de manera lógica.
- Memoria: administrarla, realizar *swapping* cuando sea necesario, comunicarse con el procesador para enviar y recibir datos e instrucciones.

Un ejemplo de llamadas al sistema para manejo de archivos usando *fcntl.h*:

```
#include <stdio.h>
#include <stdlib.h>
#include <fcntl.h>

// Permite leer un archivo y determinar el número de bytes leídos.
int main() {
    int fd, sz;
    char c[1000];

    fd = open("./archivo.txt", O_RDONLY);
    if (fd < 0) {
        perror("Error al leer el archivo");
        exit(1);
    }
    // Le pasamos al arreglo c los caracteres que se leyeron del archivo
    sz = read(fd, c, 1000);
    c[sz] = '\0';
    printf("\nLa llamada al sistema indica que se leyeron %d bytes", sz);
    printf("\nEl texto leído es: %s", c);
}
```

La librería `fcntl` manipula el descriptor (file descriptor – `fd`) de un archivo abierto.

La variable `fd` es el descriptor del archivo obtenido con la instrucción `open`.

El arreglo de caracteres `c` es un buffer con tamaño 1000 Bytes o caracteres, donde se almacenarán los caracteres leídos desde el archivo.

El tercer argumento de `read` es el número máximo de Bytes que pueden ser leídos. En este caso 1000.

La instrucción `c[sz] = '\0'` solo permite añadir un valor de fin de cadena. Si este no existe la parte final se añadirá caracteres basura al final del arreglo.